

NNN		NNN	MMM	MMM	LLL
NNN		NNN	MMM	MMM	LLL
NNN		NNN	MMM	MMM	LLL
NNN		NNN	MMMMMM	MMMMMM	LLL
NNN		NNN	MMMMMM	MMMMMM	LLL
NNN		NNN	MMMMMM	MMMMMM	LLL
NNNNNN		NNN	MMM	MMM	LLL
NNNNNN		NNN	MMM	MMM	LLL
NNNNNN		NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNNNNN	NNN	MMM	MMM	LLL
NNN	NNNNNN	NNN	MMM	MMM	LLL
NNN	NNNNNN	NNN	MMM	MMM	LLL
NNN	NNNNNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLL
NNN	NNN	NNN	MMM	MMM	LLLLLLLLLLLLLLLL
NNN	NNN	NNN	MMM	MMM	LLLLLLLLLLLLLLLL
NNN	NNN	NNN	MMM	MMM	LLLLLLLLLLLLLLLL

_S

Ps

NP

NP

\$G

\$O

NP

PA

_L

NN	NN	MM	MM	LL	PPPPPPPP	MM	MM	AAAAAA	NN	NN	IIIIII	PPPPPPPP	
NN	NN	MM	MM	LL	PPPPPPPP	MM	MM	AAAAAA	NN	NN	IIIIII	PPPPPPPP	
NN	NN	MMM	MMM	LL	PP	PP	MMM	AA	NN	NN	II	PP	PP
NN	NN	MMM	MMM	LL	PP	PP	MMM	AA	NN	NN	II	PP	PP
NNNN	NN	MM	MM	LL	PP	PP	MM	AA	NNNN	NN	II	PP	PP
NNNN	NN	MM	MM	LL	PP	PP	MM	AA	NNNN	NN	II	PP	PP
NN	NN	MM	MM	LL	PPPPPPPP	MM	MM	AA	NN	NN	II	PPPPPPPP	
NN	NN	MM	MM	LL	PPPPPPPP	MM	MM	AA	NN	NN	II	PPPPPPPP	
NN	NNNN	MM	MM	LL	PP	MM	MM	AAAAAAAA	NN	NNNN	II	PP	
NN	NNNN	MM	MM	LL	PP	MM	MM	AAAAAAAA	NN	NNNN	II	PP	
NN	NN	MM	MM	LL	PP	MM	MM	AA	NN	NN	II	PP	
NN	NN	MM	MM	LL	PP	MM	MM	AA	NN	NN	II	PP	
NN	NN	MM	MM	LL	PP	MM	MM	AA	NN	NN	II	PP	
NN	NN	MM	MM	LLLLLLLLLLLL	PP	MM	MM	AA	NN	NN	IIIIII	PP	
NN	NN	MM	MM	LLLLLLLLLLLL	PP	MM	MM	AA	NN	NN	IIIIII	PP	

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```



```
0001 0 %TITLE 'NML internal parameter manipulation module'
0002 0 MODULE NML$PMANIP (
0003 0     LANGUAGE (BLISS32),
0004 0     ADDRESSING_MODE (NONEXTERNAL=GENERAL),
0005 0     ADDRESSING_MODE (EXTERNAL=GENERAL),
0006 0     IDENT = 'V04-000'
0007 0 ) =
0008 1 BEGIN
0009 1
0010 1 *****
0011 1 *
0012 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0013 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0014 1 *   ALL RIGHTS RESERVED.
0015 1 *
0016 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0017 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0018 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0019 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0020 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0021 1 *   TRANSFERRED.
0022 1 *
0023 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0024 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0025 1 *   CORPORATION.
0026 1 *
0027 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0028 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0029 1 *
0030 1 *
0031 1 *****
0032 1
0033 1
0034 1 ABSTRACT:
0035 1
0036 1     This module contains routines to handle internal parameter
0037 1     manipulation functions.
0038 1
0039 1 ENVIRONMENT: VAX/VMS Operating System
0040 1
0041 1 AUTHOR: Distributed Systems Software Engineering
0042 1
0043 1 CREATION DATE: 23-JAN-1980
0044 1
0045 1 MODIFIED BY:
0046 1
0047 1     V03-003 MKP0003      Kathy Perko      4-Aug-1983
0048 1     Make permanent database routines transparent to the length
0049 1     of the ISAM keys at the beginning of the records.
0050 1
0051 1     V03-002 MKP0002      Kathy Perko      22-June-1982
0052 1     Add support for specifying "active X25-Protocol network".
0053 1
0054 1     V03-001 MKP0001      Kathy Perko      28-April-1982
0055 1     More modifications for NETACP control Q10. Add the
0056 1     second search key to NFB. Also, delete the start key.
0057 1
```

NMLSPMANIP
V04-000

NML internal parameter manipulation module

C 12
16-Sep-1984 00:26:09
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLPMANIP.B32;1

Page 2
(1)

: 58
: 59
: 60
: 61

0058 1 :
0059 1 :
0060 1 :--
0061 1

V02-001 LMK0001 Len Kawell 21-Jul-1981
Modifications for new NETACP control Q10.

NM
VO


```

: 63      0062 1 %SBTTL 'Declarations'
: 64      0063 1
: 65      0064 1
: 66      0065 1 : TABLE OF CONTENTS:
: 67      0066 1
: 68      0067 1
: 69      0068 1 FORWARD ROUTINE
: 70      0069 1     NML$SAVEPARAM,
: 71      0070 1     NML$CHKPRMVAL,
: 72      0071 1     NML$BLDSETQBF      : NOVALUE,
: 73      0072 1     NML$DEL_FIELDS,
: 74      0073 1     NML$ADD_FIELDS,
: 75      0074 1     NML$READPARLIST,
: 76      0075 1     NML$SHOWPARLIST,
: 77      0076 1     NML$BLDALLDES;
: 78      0077 1
: 79      0078 1
: 80      0079 1 : INCLUDE FILES:
: 81      0080 1
: 82      0081 1
: 83      0082 1 LIBRARY 'LIB$:NMLLIB.L32';
: 84      0083 1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
: 85      0084 1 LIBRARY 'SHRLIB$:NET.L32';
: 86      0085 1 LIBRARY 'SYSS$LIBRARY:STARLET.L32';
: 87      0086 1
: 88      0087 1
: 89      0088 1 : EXTERNAL REFERENCES:
: 90      0089 1
: 91      0090 1
: 92      0091 1 $NML_EXTDEF;
: 93      0092 1
: 94      0093 1 EXTERNAL LITERAL
: 95      0094 1     NML$_DSCBFOVF,
: 96      0095 1     NML$_QIOBFOVF,
: 97      0096 1     NML$_RECBFOVF;
: 98      0097 1
: 99      0098 1 EXTERNAL ROUTINE
: 100     0099 1     NML$SEARCHFLD,
: 101     0100 1     NML$BLD_REPLY,
: 102     0101 1     NML$BLDP2,
: 103     0102 1     NML$ERROR_1,
: 104     0103 1     NML$ERROR_2;
: 105     0104 1
```

```
107 0105 1 %SBTTL 'NML$SAVEPARAM Check parameter value'
108 0106 1 GLOBAL ROUTINE NML$SAVEPARAM (CPT_INDEX, LENGTH, POINTER) =
109 0107 1
110 0108 1 !++
111 0109 1 FUNCTIONAL DESCRIPTION:
112 0110 1
113 0111 1 This routine saves a parameter as a descriptor in the parameter
114 0112 1 descriptor block.
115 0113 1
116 0114 1 FORMAL PARAMETERS:
117 0115 1
118 0116 1 CPT_INDEX
119 0117 1 LENGTH
120 0118 1 POINTER
121 0119 1
122 0120 1 IMPLICIT INPUTS:
123 0121 1
124 0122 1 NML$AB_PRMSEM is the parameter semantic table.
125 0123 1 NML$AW_PRM_DES is the parameter descriptor buffer.
126 0124 1 NML$GW_PRMDESCNT contains the current number of descriptor entries.
127 0125 1
128 0126 1 IMPLICIT OUTPUTS:
129 0127 1
130 0128 1 If the parameter is valid then a descriptor entry will be created for
131 0129 1 it in NML$AW_PRM_DES and NML$GW_PRMDESCNT will be incremented.
132 0130 1
133 0131 1 ROUTINE VALUE:
134 0132 1 COMPLETION CODES:
135 0133 1
136 0134 1 Always returns NML$STS_SUC.
137 0135 1
138 0136 1 SIDE EFFECTS:
139 0137 1
140 0138 1 If the parameter descriptor buffer is full then a software error
141 0139 1 (NML$C_STS_MPR) is signalled with optional text to identify the error.
142 0140 1
143 0141 1 --
144 0142 1
145 0143 2 BEGIN
146 0144 2
147 0145 2 BIND
148 0146 2 CPT_LIST = NML$AB_CPTABLE [CPT_INDEX, 0,0,0,0]
149 0147 2 : BBLOCK [CPT$K_ENTRYLEN],
150 0148 2 SEMANTIC_LIST = NML$AB_PRMSEM [CPT_LIST [CPT$W_PSTINDEX], 0,0,0,0]
151 0149 2 : BBLOCK [PST$K_ENTRYLEN];
152 0150 2
153 0151 2 LOCAL
154 0152 2 VEC_INDEX,
155 0153 2 MASK : BLOCK [1, WORD],
156 0154 2 MSGSIZE, : Resultant message size
157 0155 2 OFFSET, : Temporary parameter offset
158 0156 2 VEC : REF BLOCKVECTOR [, 2, WORD];
159 0157 2
160 0158 2
161 0159 2
162 0160 2 Check the parameter descriptor buffer to see if there is any room left
163 0161 2
```



```
: 164      0162      2      IF .NML$GW_PRMDSCNT GEQU PDB$K_NUMBER
: 165      0163      2      THEN
: 166      0164      2      BEGIN
: 167      0165      2      |
: 168      0166      2      | Signal parameter descriptor buffer overflow.
: 169      0167      2      |
: 170      0168      2      |     NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSB$M_DET_FLD OR MSB$M_MSG_FLD;
: 171      0169      2      |     NML$AB_MSGBLOCK [MSB$B_CODE] = NML$C_STS_MPR; ! Get error code
: 172      0170      2      |     NML$AB_MSGBLOCK [MSB$W_DETAIL] =
: 173      0171      2      |     .SEMANTIC_LIST [PST$W_DATAID]; ! Get parameter code detail
: 174      0172      2      |     NML$AB_MSGBLOCK [MSB$L_TEXT] = NML$DSCBFOVF;
: 175      0173      2      |     NML$BLD_REPLY (NML$AB_MSGBLOCK, MSGSIZE); ! Build message
: 176      0174      2      |     $SIGNAL_MSG (NML$AB_SNDBUFFER, .MSGSIZE); ! Signal error message
: 177      0175      2      |
: 178      0176      2      |     END;
: 179      0177      2      |
: 180      0178      2      | Add descriptor entry for this parameter.
: 181      0179      2      |
: 182      0180      2      |     NML$AW_PRM_DES [.NML$GW_PRMDSCNT, PDB$W_INDEX] = .CPT_INDEX;
: 183      0181      2      |     NML$AW_PRM_DES [.NML$GW_PRMDSCNT, PDB$W_COUNT] = .LENGTH;
: 184      0182      2      |     NML$AW_PRM_DES [.NML$GW_PRMDSCNT, PDB$A_POINTER] = .POINTER;
: 185      0183      2      |
: 186      0184      2      |     NML$GW_PRMDSCNT = .NML$GW_PRMDSCNT + 1; ! Increment descriptor count
: 187      0185      2      |
: 188      0186      2      |     RETURN NML$STS_SUC
: 189      0187      2      |
: 190      0188      1      |     END;
:                               | End of NML$SAVEPARAM
```

```
.TITLE NML$PMANIP NML internal parameter manipulation
      module
```

```
.IDENT \V04-000\
```

```
.EXTRN NML$GB_EVTSRCTYP
.EXTRN NML$GQ_EVTSRCDSK
.EXTRN NML$GW_EVTCLASS
.EXTRN NML$GB_EVTMSKTYP
.EXTRN NML$GQ_EVTMSKDSK
.EXTRN NML$GW_EVTSNKADR
.EXTRN NML$GW_ACP_CHAN
.EXTRN NML$GL_LOGMASK, NML$GQ_ENTSTRDSC
.EXTRN NML$AB_QIOBUFFER
.EXTRN NML$GQ_QIOBFDSC
.EXTRN NML$AB_EXEBUFFER
.EXTRN NML$GL_EXEDATPTR
.EXTRN NML$GQ_EXEDATDSC
.EXTRN NML$GQ_EXEBFDSC
.EXTRN NML$AB_RCVBUFFER
.EXTRN NML$GQ_RCVBFDSC
.EXTRN NML$AB_SNDBUFFER
.EXTRN NML$GQ_SNDBFDSC
.EXTRN NML$GL_RCVDATLEN
.EXTRN NML$AB_CPTABLE, NML$AB_MSGBLOCK
.EXTRN NML$AB_ENTITY_ID
.EXTRN NML$AB_QUALIFIER_ID
.EXTRN NML$AB_ENTITYDATA
.EXTRN NML$AB_NML_NMV, NML$AB_PRMSEM
```

```
50      04      AC      00000000G0040 001C 00000
54      00000000G 00 9E 00002
53      00000000G 00 9E 00009
52      00000000G 00 9E 00010
5E      04      C2 00017
04      AC      0A      C5 0001A
00000000G0040 9F 0001F
50      9E      3C 00026
50      10      C4 00029
20      63      B1 0002C
3A      1F 0002F
62      06      D0 00031
04      A2      05      8E 00034
00000000G0040 9F 00038
08      A2      9E      B0 0003F
0C      A2      00000000G 8F D0 00043
4004      8F BB 0004B
00000000G 00 02 FB 0004F
00000000G 00 6E DD 00056
00000000G 01F90000 00 9F 00058
00000000G 00 03 FB 00064
50      63      3C 0006B 1$:
9E      04      AC      B0 00071
02      A440 7F 00075
9E      08      AC      B0 00079
04      A440 7F 0007D
9E      0C      AC      D0 00081
63      B6 00085
50      01      D0 00087
04      0008A
```

```
.EXTRN NML$AB_RECBUF, NML$AL_ENTINFTAB
.EXTRN NML$AL_PERMINFTAB
.EXTRN NML$AW_PRM_DES, NML$GB_CMD_VER
.EXTRN NML$GB_ENTITY_CODE
.EXTRN NML$GB_ENTITY_FORMAT
.EXTRN NML$GL_QUALIFIER_PST
.EXTRN NML$GB_QUALIFIER_FORMAT
.EXTRN NML$GB_FUNCTION
.EXTRN NML$GB_INFO, NML$GB_OPTIONS
.EXTRN NML$GL_PRMCODE, NML$GL_PRS_FLGS
.EXTRN NML$GL_NML_ENTITY
.EXTRN NML$GQ_NETNAMDSC
.EXTRN NML$GQ_RECBFDSC
.EXTRN NML$GW_PRMDSCNT
.EXTRN NML$DSCBFOVF, NML$QIOBFOVF
.EXTRN NML$RECBFOVF, NML$SEARCHFLD
.EXTRN NML$BLD_REPLY, NML$BLDP2
.EXTRN NML$ERROR_1, NML$ERROR_2
```

.PSECT \$CODE\$,NOWRT,2

```
.ENTRY NML$SAVEPARAM, Save R2,R3,R4 ; 0106
MOVAB NML$AW_PRM_DES, R4 ;
MOVAB NML$GW_PRMDSCNT, R3 ;
MOVAB NML$AB_MSGBLOCK, R2 ;
SUBL2 #4, SP ;
MULL3 #10, CPT_INDEX, R0 ; 0146
PUSHAB NML$AB_CPTABLE[R0] ; 0148
MOVZWL @ (SP)+, R0 ;
MULL2 #16, R0 ;
CMPW NML$GW_PRMDSCNT, #32 ; 0162
BLSSU 1$ ;
MOVL #6, NML$AB_MSGBLOCK ; 0168
MNEGB #5, NML$AB_MSGBLOCK+4 ; 0169
PUSHAB NML$AB_PRMSEM[R0] ; 0171
MOVW @ (SP)+, NML$AB_MSGBLOCK+8 ;
MOVL #NML$DSCBFOVF, NML$AB_MSGBLOCK+12 ; 0172
PUSHR #^M<R2, SP> ; 0173
CALLS #2, NML$BLD_REPLY ;
PUSHL MSGSIZE ; 0174
PUSHAB NML$AB_SNDBUFFER ;
PUSHL #33095880 ;
CALLS #3, LIB$SIGNAL ;
MOVZWL NML$GW_PRMDSCNT, R0 ; 0180
PUSHAQ NML$AW_PRM_DES[R0] ;
MOVW CPT_INDEX, @ (SP)+ ;
PUSHAQ NML$AW_PRM_DES+2[R0] ; 0181
MOVW LENGTH, @ (SP)+ ;
PUSHAQ NML$AW_PRM_DES+4[R0] ; 0182
MOVL POINTER, @ (SP)+ ;
INCW NML$GW_PRMDSCNT ; 0184
MOVL #1, R0 ; 0186
RET ; 0188
```

; Routine Size: 139 bytes, Routine Base: \$CODE\$ + 0000


```
.. 192 0189 1 %SBTTL 'NML$CHKPRMVAL Check parameter value'
.. 193 0190 1 GLOBAL ROUTINE NML$CHKPRMVAL (CPT_INDEX, LEN, ADR) =
.. 194 0191 1
.. 195 0192 1 ++
.. 196 0193 1 FUNCTIONAL DESCRIPTION:
.. 197 0194 1
.. 198 0195 1 This routine verifies that parameter values from the NICE message
.. 199 0196 1 fall within valid boundaries.
.. 200 0197 1
.. 201 0198 1 FORMAL PARAMETERS:
.. 202 0199 1
.. 203 0200 1 CPT_INDEX Index into change parameter table.
.. 204 0201 1 LEN Byte count of parameter.
.. 205 0202 1 ADR Address of parameter.
.. 206 0203 1
.. 207 0204 1 IMPLICIT INPUTS:
.. 208 0205 1
.. 209 0206 1 NONE
.. 210 0207 1
.. 211 0208 1 IMPLICIT OUTPUTS:
.. 212 0209 1
.. 213 0210 1 NONE
.. 214 0211 1
.. 215 0212 1 ROUTINE VALUE:
.. 216 0213 1 COMPLETION CODES:
.. 217 0214 1
.. 218 0215 1 Returns success (NML$_STS_SUC) if the paramter value is within range.
.. 219 0216 1
.. 220 0217 1 SIDE EFFECTS:
.. 221 0218 1
.. 222 0219 1 An error message (NMA$C_STS_PVA) is signalled if the value is bad.
.. 223 0220 1
.. 224 0221 1 --
.. 225 0222 1
.. 226 0223 2 BEGIN
.. 227 0224 2
.. 228 0225 2 LOCAL
.. 229 0226 2 MAX, ! Maximum parameter value (0 if no limit)
.. 230 0227 2 MIN, ! Minimum parameter value
.. 231 0228 2 VAL, ! Parameter value to compare
.. 232 0229 2 STATUS; ! Status of the range checking operations
.. 233 0230 2
.. 234 0231 2 The parameter semantic table index is determined by looking in the change
.. 235 0232 2 parameter table.
.. 236 0233 2
.. 237 0234 2 BIND
.. 238 0235 2 CPT_LIST = NML$AB CPTABLE [.CPT_INDEX, 0,0,0,0]
.. 239 0236 2 : BBLOCK [CPT$K_ENTRYLEN]
.. 240 0237 2 SEMANTIC_LIST = NML$AB PRMSEM [.CPT_LIST [CPT$W_PSTINDEX], 0,0,0,0]
.. 241 0238 2 : BBLOCK [PST$K_ENTRYLEN];
.. 242 0239 2
.. 243 0240 2 Pick up the values for comparison.
.. 244 0241 2
.. 245 0242 2 MIN = .SEMANTIC_LIST [PST$L_MINVALUE];
.. 246 0243 2 MAX = .SEMANTIC_LIST [PST$L_MAXVALUE];
.. 247 0244 2 STATUS = NML$_STS_SUC;
.. 248 0245 2 !
```

```
249 0246 2 ! If the parameter is a string then get the byte count (a byte). If the
250 0247 2 ! parameter is not a string then get the value of the appropriate width
251 0248 2 ! (byte, word, longword).
252 0249 2 !
253 0250 2 ! IF .SEMANTIC_LIST [PST$B_FORMAT] EQLU NML$K_STRING
254 0251 2 ! THEN
255 0252 2 !     VAL = .LEN
256 0253 2 ! ELSE
257 0254 2 !     VAL = .(.ADR)<0,.LEN*8>;
258 0255 2 !
259 0256 2 ! Check the minimum parameter value.
260 0257 2 !
261 0258 2 ! IF .VAL LSSU .MIN
262 0259 2 ! THEN
263 0260 2 !     STATUS = NML$_STS_PVA;
264 0261 2 !
265 0262 2 ! If the maximum value has a zero in it then don't bother to check it.
266 0263 2 !
267 0264 2 ! IF .MAX NEQU 0
268 0265 2 ! AND .VAL GTRU .MAX
269 0266 2 ! THEN
270 0267 2 !     STATUS = NML$_STS_PVA;
271 0268 2 !
272 0269 2 ! If the parameter is not within range then signal a parameter value error.
273 0270 2 !
274 0271 2 ! IF NOT .STATUS
275 0272 2 ! THEN
276 0273 2 !     NML$ERROR_2 (NML$_STS_PVA, .SEMANTIC_LIST [PST$W_DATAID]);
277 0274 2 !
278 0275 2 ! RETURN NML$_STS_SUC
279 0276 2 !
280 0277 1 ! END;                                     ! End of NML$CHKPRMVAL
```

50	04	AC	00000000G0040	0A C5 00002	.ENTRY NML\$CHKPRMVAL, Save R2,R3,R4,R5	0190
				9E 3C 00007	MULL3 #10, CPT_INDEX, R0	0235
				10 C4 00011	PUSHAB NML\$AB_CPTABLE[R0]	0237
				9E 00014	MOVZWL @ (SP)+, R0	
				A1 D0 0001C	MULL2 #16, R0	
				A1 D0 00020	MOVAB NML\$AB_PRMSEM[R0], R1	
				01 D0 00024	MOVL 4(R1), MIN	0242
				A1 91 00027	MOVL 8(R1), MAX	0243
				06 12 0002B	MOVL #1, STATUS	0244
				AC D0 0002D	CMPB 2(R1), #3	0250
				0B 11 00031	BNEQ 1\$	
				03 78 00033	MOVL LEN, VAL	0252
				00 EF 00038	BRB 2\$	
52	0C	50	08	52 D1 0003E	ASHL #3, LEN, R0	0254
				03 1E 00041	EXTZV #0, R0, @ADR, VAL	
				20 CE 00043	CMPB VAL, MIN	0258
				54 D5 00046	BGEQU 3\$	
				08 13 00048	MNEGL #32, STATUS	0260
					TSTL MAX	0264
					BEQL 4\$	

NML\$PMANIP
V04-000

NML internal parameter manipulation module
NML\$CHKPRMVAL Check parameter value

J 12
16-Sep-1984 00:26:09
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLPMANIP.B32;1

Page 9
(4)

54	52	D1	0004A	CMPL	VAL, MAX	: 0265
	03	1B	0004D	BLEQU	4\$: 0267
53	20	CE	0004F	MNEGL	#32, STATUS	: 0271
0D	53	E8	00052	BLBS	STATUS, 5\$: 0273
7E	61	3C	00055	MOVZWL	(R1), -(SP)	: 0275
7E	10	CE	00058	MNEGL	#16, -(SP)	: 0277
00000000G 00	02	FB	0005B	CALLS	#2, NML\$ERROR_2	
50	01	D0	00062	MOVL	#1, R0	
	04	00065	RET			

; Routine Size: 102 bytes, Routine Base: \$CODE\$ + 008B

```
282 0278 1 %SBTTL 'NML$BLDSETQBF Build SET/CLEAR QIO buffers'
283 0279 1 GLOBAL ROUTINE NML$BLDSETQBF (FUNC, DB,
284 0280 1 KEYID1, KEYLEN1, KEYADR1,
285 0281 1 KEYID2, KEYLEN2, KEYADR2,
286 0282 1 NFB$FDSC, NFB$DSC,
287 0283 1 P2B$FDSC, P2DSC,
288 0284 1 VALB$FDSC, VALDSC)
289 0285 1 : NOVALUE =
290 0286 1
291 0287 1 ++
292 0288 1 FUNCTIONAL DESCRIPTION:
293 0289 1
294 0290 1 This routine builds a QIO buffer for parameter modifications.
295 0291 1
296 0292 1 FORMAL PARAMETERS:
297 0293 1
298 0294 1 FUNC Control function - NFB$C_FC_SET or NFB$C_FC_CLEAR
299 0295 1 DB Database ID
300 0296 1 KEYID1 Search key one ID
301 0297 1 KEYLEN1 Search key one length
302 0298 1 KEYADR1 Search key one address
303 0299 1 KEYID2 Search key two ID
304 0300 1 KEYLEN2 Search key two length
305 0301 1 KEYADR2 Search key two address
306 0302 1 NFB$FDSC Descriptor of control function buffer (P1)
307 0303 1 NFB$DSC Descriptor of resulting control function buffer (P1)
308 0304 1 P2B$FDSC Descriptor of P2 buffer
309 0305 1 P2DSC Descriptor of resulting P2 buffer
310 0306 1 VALB$FDSC Descriptor of parameter value buffer (P4)
311 0307 1 VALDSC Descriptor of resulting parameter value data (P4)
312 0308 1
313 0309 1 NML$AW_PRM DES List of parameter descriptors
314 0310 1 NML$GW_PRMDESCNT Count of parameter descriptors
315 0311 1
316 0312 1 OUTPUTS:
317 0313 1
318 0314 1 -- Output buffers and descriptors built.
319 0315 1
320 0316 1
321 0317 2 BEGIN
322 0318 2
323 0319 2 MAP
324 0320 2 NFB$FDSC : REF DESCRIPTOR,
325 0321 2 NFB$DSC : REF DESCRIPTOR,
326 0322 2 P2B$FDSC : REF DESCRIPTOR,
327 0323 2 P2DSC : REF DESCRIPTOR,
328 0324 2 VALB$FDSC : REF DESCRIPTOR,
329 0325 2 VALDSC : REF DESCRIPTOR;
330 0326 2
331 0327 2 LOCAL
332 0328 2 MSGSIZE,
333 0329 2 CPT_INDEX,
334 0330 2 NFB : REF BBLOCK [NFB$C_LENGTH],
335 0331 2 VALADR,
336 0332 2 VALPTR,
337 0333 2 VALLEN,
338 0334 2 VALTYP,
```



```
339 0335 2 CPT: REF BBLOCK [CPT$K_ENTRYLEN],
340 0336 2 PST: REF BBLOCK [PST$K_ENTRYLEN];
341 0337 2
342 0338 2 BIND
343 0339 2 VALBUF = VALBFDSC [DSC$A_POINTER] : REF BBLOCK;
344 0340 2
345 0341 2
346 0342 2
347 0343 2
348 0344 2
349 0345 2
350 0346 2 NFB DSC [DSC$A_POINTER] = NFB = .NFB BFDSC [DSC$A_POINTER];
351 0347 2 CH$FILL(0, $BYTEOFFSET(NFB$SL_FLDID), .NFB); ! Zero NFB header
352 0348 2 NFB [NFB$B_FCT] = .FUNC;
353 0349 2 NFB [NFB$B_DATABASE] = .DB;
354 0350 2 NFB [NFB$SL_SRCH_KEY] = .KEYID1;
355 0351 2 NFB [NFB$SL_SRCH2_KEY] = .KEYID2;
356 0352 2 NFB = NFB[NFB$SL_FLDID];
357 0353 2
358 0354 2
359 0355 2 Build the P2 buffer. The P2 buffer contains the values of the search
360 0356 2 keys. The search keys are used by NETACP to identify the database entry
361 0357 2 to be updated.
362 0358 2
363 0359 2 SELECTONEU .KEYID1 OF
364 0360 2 SET
365 0361 2 [NFB$C_EFI_SIN]: ! Logging filters (sink node)
366 0362 2 NML$BLDP2 (0, .(.KEYADR1)<0,16>, -1, 0, .P2BFDSC, .P2DSC);
367 0363 2
368 0364 2 [NFB$C_ESI_SNK]: ! Logging sink
369 0365 2 NML$BLDP2 (0, .(.KEYADR1)<0,8>, -1, 0, .P2BFDSC, .P2DSC);
370 0366 2
371 0367 2 [NFB$C_NDI_ADD]: ! Node (by address)
372 0368 2 NML$BLDP2 (0, .(.KEYADR1)<0,16>, -1, 0, .P2BFDSC, .P2DSC);
373 0369 2
374 0370 2 [NFB$C_XGI_GRP]: ! Protocol Groups.
375 0371 2
376 0372 2 Protocol Group is always qualified with a DTE ID. The DTE ID
377 0373 2 is the second search key.
378 0374 2
379 0375 2 NML$BLDP2 (.KEYLEN1, .KEYADR1, .KEYLEN2, .KEYADR2,
380 0376 2 .P2BFDSC, .P2DSC);
381 0377 2
382 0378 2 [NFB$C_XNI_NET]:
383 0379 2 BEGIN
384 0380 2
385 0381 2 If search key 1 length is zero, then the QIO is for the "active
386 0382 2 X-25 Protocol network". Pass NML$BLDP2 the code to indicate that
387 0383 2 the key value is a word.
388 0384 2
389 0385 2 IF .KEYLEN1 EQL 0 THEN
390 0386 2 KEYLEN1 = -2;
391 0387 2 NML$BLDP2 (.KEYLEN1, .KEYADR1, -1, 0, .P2BFDSC, .P2DSC);
392 0388 2 END;
393 0389 2
394 0390 2 [OTHERWISE]:
395 0391 2 NML$BLDP2 (.KEYLEN1, .KEYADR1, -1, 0, .P2BFDSC, .P2DSC);
```

```

396 0392 2
397 0393 2
398 0394 2
399 0395 2
400 0396 2
401 0397 2
402 0398 2
403 0399 2
404 0400 2
405 0401 2
406 0402 2
407 0403 2
408 0404 2
409 0405 2
410 0406 2
411 0407 2
412 0408 2
413 0409 2
414 0410 2
415 0411 2
416 0412 2
417 0413 2
418 0414 2
419 0415 2
420 0416 2
421 0417 2
422 0418 2
423 0419 2
424 0420 2
425 0421 2
426 0422 2
427 0423 2
428 0424 2
429 0425 2
430 0426 2
431 0427 2
432 0428 2
433 0429 2
434 0430 2
435 0431 2
436 0432 2
437 0433 2
438 0434 2
439 0435 2
440 0436 2
441 0437 2
442 0438 2
443 0439 2
444 0440 2
445 0441 2
446 0442 2
447 0443 2
448 0444 2
449 0445 2
450 0446 2
451 0447 2
452 0448 2

TES;

Setup parameter value buffer descriptor

VALDSC [DSC$A_POINTER] = VALPTR = .VALBFDSC [DSC$A_POINTER];

For each entry in the parameter descriptor list, add its ACP identifier
to the NFB and its value to the value buffer.

INCR I FROM 0 TO .NML$GW_PRMDSCNT - 1 DO
BEGIN
  CPT_INDEX = .NML$AW_PRM_DES [I, PDB$W_INDEX];
  CPT = NML$AB_CPTABLE [CPT_INDEX, 0,0,0,0];
  PST = NML$AB_PRMSEM [CPT [CPT$W_PST_INDEX], 0,0,0,0];
  VALLEN = .NML$AW_PRM_DES [I, PDB$W_COUNT];
  VALADR = .NML$AW_PRM_DES [I, PDB$A_POINTER];

  IF (.VALPTR + .VALLEN + 2 LSSU
      .VALBFDSC [DSC$A_POINTER] + .VALBFDSC [DSC$W_LENGTH]) AND
      (.NFB + 4 LSSU
      .NFBFDSC [DSC$A_POINTER] + .NFBFDSC [DSC$W_LENGTH])
  THEN
    BEGIN
      NFB[0,0,32,0] = .PST [PST$L_NFBID];
      NFB = .NFB + 4;

      IF .VALLEN GTRU 0
      THEN
        BEGIN
          VALTYP = .(PST [PST$L_NFBID])
            <$BITPOSITION (NFB$V_TYP),
            $FIELDWIDTH (NFB$V_TYP)>;

          IF .VALTYP EQLU NFB$C_TYP_STR
          THEN
            BEGIN
              (.VALPTR)<0,16> = .VALLEN;      ! Set count
              VALPTR = .VALPTR + 2;
              VALPTR = CH$MOVE (.VALLEN, .VALADR, .VALPTR);

            END
          ELSE
            BEGIN
              (.VALPTR)<0,32> = .(.VALADR)<0,.VALLEN*8>;
              VALPTR = .VALPTR + 4;      ! Increment data pointer
            END;
        END
      ELSE
        END
    END
  ELSE
    END
```



```
453 0449 4 BEGIN
454 0450 4
455 0451 4 NML$AB_MSGBLOCK [MSB$L_FLAGS] = MSB$M MSG FLD; ! Set message text flag
456 0452 4 NML$AB_MSGBLOCK [MSB$B_CODE] = NMA$C STS MPR;
457 0453 4 NML$AB_MSGBLOCK [MSB$L_TEXT] = NML$ QIOBFOVF;
458 0454 4 NML$BLD_REPLY (NML$AB_MSGBLOCK, MSGSIZE); ! Build message
459 0455 4 $SIGNAL_MSG (NML$AB_SNDBUFFER, .MSGSIZE); ! Signal it
460 0456 4
461 0457 3 END;
462 0458 3
463 0459 2 END;
464 0460 2
465 0461 2 NFB [0,0,32,0] = NFB$C ENDOFLIST;
466 0462 2 NFB$DSC [DSC$W_LENGTH] = .NFB - .NFB$DSC[DSC$A_POINTER] + 4;
467 0463 2
468 0464 2 VALDSC [DSC$W_LENGTH] = .VALPTR - .VALDSC[DSC$A_POINTER];
469 0465 2
470 0466 1 END; ! End of NML$BLDSETQBF
```

				OFFC 00000	.ENTRY	NML\$BLDSETQBF, Save R2,R3,R4,R5,R6,R7,R8,- R9,R10,R11	
		5E		14 C2 00002	SUBL2	#20, SP	0279
			34	AC DD 00005	PUSHL	VALBFDSC	0339
50	24	AC		04 C1 00008	ADDL3	#4, NFBFDSC, R0	0346
		56		60 D0 0000D	MOVL	(R0), NFB	
51	28	AC		04 C1 00010	ADDL3	#4, NFB\$DSC, R1	
		61		56 D0 00015	MOVL	NFB, (R1)	
10	00	6E		00 2C 00018	MOVC5	#0, (SP), #0, #16, (NFB)	0347
				66 0001D			
		86	04	AC 90 0001E	MOVB	FUNC, (NFB)+	0348
	01	A6	08	AC 90 00022	MOVB	DB, 1(NFB)	0349
		50	0C	AC D0 00027	MOVL	KEYID1, R0	0350
	03	A6		50 D0 0002B	MOVL	R0, 3(NFB)	
	07	A6	18	AC D0 0002F	MOVL	KEYID2, 7(NFB)	0351
		56		0F C0 00034	ADDL2	#15, NFB	0352
06010010		8F		50 D1 00037	CMPL	R0, #100728848	0361
				21 13 0003E	BEQL	2\$	
07010010		8F		50 D1 00040	CMPL	R0, #117506064	0364
				0F 12 00047	BNEQ	1\$	
		7E	2C	AC 7D 00049	MOVQ	P2BFDSC, -(SP)	0365
		7E		7E D4 0004D	CLRL	-(SP)	
		7E		01 CE 0004F	MNEGL	#1, -(SP)	
		7E	14	BC 9A 00052	MOVZBL	@KEYADR1, -(SP)	
				16 11 00056	BRB	3\$	
02010012		8F		50 D1 00058 1\$:	CMPL	R0, #33619986	0367
				11 12 0005F	BNEQ	4\$	
		7E	2C	AC 7D 00061 2\$:	MOVQ	P2BFDSC, -(SP)	0368
				7E D4 00065	CLRL	-(SP)	
		7E		01 CE 00067	MNEGL	#1, -(SP)	
		7E	14	BC 3C 0006A	MOVZWL	@KEYADR1, -(SP)	
				7E D4 0006E 3\$:	CLRL	-(SP)	
				32 11 00070	BRB	8\$	
0A020041		8F		50 D1 00072 4\$:	CMPL	R0, #167903297	0370

			0A 12 00079	BNEQ 5\$		
	7E	2C	AC 7D 0007B	MOVQ P2BFDSC, -(SP)		0376
	7E	1C	AC 7D 0007F	MOVQ KEYLEN2, -(SP)		0375
			1B 11 00083	BRB 7\$		
09020041	8F		50 D1 00085 5\$:	CMPL R0, #151126081		0378
			09 12 0008C	BNEQ 6\$		
		10	AC D5 0008E	TSTL KEYLEN1		0385
			04 12 00091	BNEQ 6\$		
10	AC		02 CE 00093	MNEGL #2, KEYLEN1		0386
	7E	2C	AC 7D 00097 6\$:	MOVQ P2BFDSC, -(SP)		0391
			7E D4 0009B	CLRL -(SP)		
	7E		01 CE 0009D	MNEGL #1, -(SP)		
	7E	10	AC 7D 000A0 7\$:	MOVQ KEYLEN1, -(SP)		
00000000G	00		06 FB 000A4 8\$:	CALLS #6, NML\$BLDP2		
	5B	38	AC D0 000AB	MOVL VALDSC, R11		0399
50	6E		04 C1 000AF	ADDL3 #4, (SP), R0		
	53		60 D0 000B3	MOVL (R0), VALPTR		
04	AB		53 D0 000B6	MOVL VALPTR, 4(R11)		
0C	AE	00000000G	00 3C 000BA	MOVZWL NML\$GW_PRMDESCNT, 12(SP)		0405
	5A		01 CE 000C2	MNEGL #1, I		
		00D8	31 000C5	BRW 12\$		
		00000000G	004A 7F 000C8 9\$:	PUSHAQ NML\$AW_PRM_DESC[I]		0408
	59		9E 3C 000CF	MOVZWL @ (SP)+, CPT_INDEX		
50	59		0A C5 000D2	MULL3 #10, CPT_INDEX, R0		0409
	58	00000000G	0040 9E 000D6	MOVAB NML\$AB_CPTABLE[R0], CPT		
	50		68 3C 000DE	MOVZWL (CPT), R0		0410
	50		10 C4 000E1	MULL2 #16, R0		
04	AE	00000000G	0040 9E 000E4	MOVAB NML\$AB_PRMSEM[R0], PST		
		00000000G	004A 7F 000ED	PUSHAQ NML\$AW_PRM_DESC+2[I]		0411
	57		9E 3C 000F4	MOVZWL @ (SP)+, VALLEN		
		00000000G	004A 7F 000F7	PUSHAQ NML\$AW_PRM_DESC+4[I]		0412
08	AE		9E D0 000FE	MOVZWL @ (SP)+, VALADR		
	51	02 A743	9E 00102	MOVAB 2(VALLEN)[VALPTR], R1		0414
	52	34 BC	3C 00107	MOVZWL @VALBFDSC, R2		0415
54	6E		04 C1 0010B	ADDL3 #4, (SP), R4		
50	64		52 C1 0010F	ADDL3 R2, (R4), R0		
	50		51 D1 00113	CMPL R1, R0		
			49 1E 00116	BGEQU 11\$		
	51	04 A6	9E 00118	MOVAB 4(R6), R1		0416
	52	24 BC	3C 0011C	MOVZWL @NFBFDSC, R2		0417
54	AC		04 C1 00120	ADDL3 #4, NFBFDSC, R4		
50	64		52 C1 00125	ADDL3 R2, (R4), R0		
	50		51 D1 00129	CMPL R1, R0		
			33 1E 0012C	BGEQU 11\$		
50	04 AE		0C C1 0012E	ADDL3 #12, PST, R0		0421
	86		60 D0 00133	MOVL (R0), (NFB)+		
			57 D5 00136	TSTL VALLEN		0424
			66 13 00138	BEQL 12\$		
	50	04 AE	0E C1 0013A	ADDL3 #14, PST, R0		0427
10	AE		00 EF 0013F	EXTZV #0, #2, (R0), VALTYP		
	02	10 AE	D1 00145	CMPL VALTYP, #2		0430
			0A 12 00149	BNEQ 10\$		
	83		57 B0 0014B	MOVW VALLEN, (VALPTR)+		0434
	63	08 BE	57 28 0014E	MOVW VALLEN, @VALADR, (VALPTR)		0436
			4B 11 00153	BRB 12\$		0430
	50		03 78 00155 10\$:	ASHL #3, VALLEN, R0		0442
83	08 BE		00 EF 00159	EXTZV #0, R0, @VALADR, (VALPTR)+		

NML\$PMANIP
V04-000

NML internal parameter manipulation module
NML\$BLDSETQBF Build SET/CLEAR QIO buffers

C 13
16-Sep-1984 00:26:09
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLPMANIP.B32;1

Page 15
(5)

00000000G	00	3F	11	0015F	BRB	12\$: 0424	
00000000G	00	04	D0	00161	11\$:	MOVL	#4, NML\$AB_MSGBLOCK	: 0451
00000000G	00	05	8E	00168	MNEGB	#5, NML\$AB_MSGBLOCK+4	: 0452	
		8F	D0	0016F	MOVL	#NML\$QIOBFOVF, NML\$AB_MSGBLOCK+12	: 0453	
		AE	9F	0017A	PUSHAB	MSGSIZE	: 0454	
		00	9F	0017D	PUSHAB	NML\$AB_MSGBLOCK		
00000000G	00	02	FB	00183	CALLS	#2, NML\$BLD_REPLY		
		AE	DD	0018A	PUSHL	MSGSIZE	: 0455	
		00	9F	0018D	PUSHAB	NML\$AB_SNDBUFFER		
		8F	DD	00193	PUSHL	#33095680		
00000000G	00	03	FB	00199	CALLS	#3, LIB\$SIGNAL		
02	5A	0C	AE	F2	001A0	12\$:	: 0405	
			03	11	001A5	BRB	14\$	
		FF	1E	31	001A7	13\$:		
			66	D4	001AA	14\$:		
50	28	AC	04	C1	001AC	CLRL	(NFB)	: 0461
		56	60	C2	001B1	ADDL3	#4, NFB DSC, R0	: 0462
28	BC	56	04	A1	001B4	SUBL2	(R0), R6	
6B	53	04	AB	A3	001B9	ADDW3	#4, R6, @NFB DSC	
			04	001BE	SUBW3	4(R11), VALPTR, (R11)	: 0464	
					RET		: 0466	

; Routine Size: 447 bytes, Routine Base: \$CODE\$ + 00F1

; 471 0467 1

```
473 0468 1 %SBTTL 'NML$ADD_FIELDS Add parameter fields to record'
474 0469 1 GLOBAL ROUTINE NML$ADD_FIELDS (BUFSIZE, RTNDSC) =
475 0470 1
476 0471 1 ++
477 0472 1 FUNCTIONAL DESCRIPTION:
478 0473 1
479 0474 1 This routine adds fields to a permanent data base record.
480 0475 1
481 0476 1 FORMAL PARAMETERS:
482 0477 1
483 0478 1 BUFSIZE Maximum size of the record buffer.
484 0479 1 RTNDSC Address of the current record descriptor.
485 0480 1
486 0481 1 IMPLICIT INPUTS:
487 0482 1
488 0483 1 NONE
489 0484 1
490 0485 1 IMPLICIT OUTPUTS:
491 0486 1
492 0487 1 The record descriptor pointed to by RTNDSC is updated to include
493 0488 1 any fields added to the record.
494 0489 1
495 0490 1 ROUTINE VALUE:
496 0491 1 COMPLETION CODES:
497 0492 1
498 0493 1 NONE
499 0494 1
500 0495 1 SIDE EFFECTS:
501 0496 1
502 0497 1 NONE
503 0498 1
504 0499 1 --
505 0500 1
506 0501 2 BEGIN
507 0502 2
508 0503 2 LOCAL
509 0504 2 CPT_INDEX, ! Change parameter table index
510 0505 2 SEM_INDEX, ! Semantic table index
511 0506 2 FLDLEN, ! Field length
512 0507 2 FLDADR, ! Field address
513 0508 2 MSGSIZE, ! Message size
514 0509 2 ROUTINE_ADR, ! Temporary routine address
515 0510 2 STATUS;
516 0511 2
517 0512 2 INCR I FROM 0 TO .NML$GW_PRMDESCNT - 1 DO
518 0513 3 BEGIN
519 0514 3
520 0515 3 FLDLEN = .NML$AW_PRM_DES [.I, PDB$W_COUNT];
521 0516 3 FLDADR = .NML$AW_PRM_DES [.I, PDB$A_POINTER];
522 0517 3
523 0518 3 CPT_INDEX = .NML$AW_PRM_DES [.I, PDB$W_INDEX];
524 0519 3 ROUTINE_ADR = .NML$AB_CPTABLE [.CPT_INDEX, CPT$A_DEFINE_RTN];
525 0520 3 SEM_INDEX = .NML$AB_CPTABLE [.CPT_INDEX, CPT$W_PSTINDEX];
526 0521 3
527 0522 4 IF NOT (STATUS =
528 0523 4 (.ROUTINE_ADR) (NML$AB_PRMSEM [.SEM_INDEX, 0,0,0,0].
529 0524 4 .BUFSIZE,
```



```
.. 530      0525  4      .FLDLEN,  
.. 531      0526  4      .FLDADR,  
.. 532      0527  4      .RTNDSC))  
.. 533      0528      THEN  
.. 534      0529      RETURN .STATUS  
.. 535      0530      END;  
.. 536      0531      RETURN NML$_STS_SUC  
.. 537      0532      END;  
.. 538      0533      ! End of NML$ADD_FIELDS  
.. 539      0534      END;  
.. 540      0535      1
```

```
                                03FC 00000  
59 00000000G 00 9E 00002      .ENTRY NML$ADD_FIELDS, Save R2,R3,R4,R5,R6,R7,R8,- : 0469  
55 00000000G 00 3C 00009      R9  
54      01 CE 00010      MOVAB NML$AW_PRM DES+2, R9  
      47 11 00013      MOVZWL NML$GW_PRMDESCNT, R5  
      6944 7F 00015 1$:      MNEGL #1, I  
58      9E 3C 00018      BRB 2$  
      02 A944 7F 0001B      PUSHAB NML$AW_PRM DES+2[I]  
57      9E D0 0001F      MOVZWL @ (SP)+, FLDLEN  
      FE A944 7F 00022      PUSHAB NML$AW_PRM DES+4[I]  
53      9E 3C 00026      MOVL @ (SP)+, FLDADR  
53      0A C5 00029      PUSHAB NML$AW_PRM DES[I]  
      00000000G0041 9F 0002D      MOVZWL @ (SP)+, CPT_INDEX  
56      9E D0 00034      MULL3 #10, CPT_INDEX, R1  
      00000000G0041 9F 00037      PUSHAB NML$AB_CPTABLE+2[R1]  
52      9E 3C 0003E      MOVL @ (SP)+, ROUTINE_ADR  
      08 AC DD 00041      PUSHAB NML$AB_CPTABLE[R1]  
      57 DD 00044      MOVZWL @ (SP)+, SEM_INDEX  
      58 DD 00046      PUSHL RTNDSC  
      04 AC DD 00048      PUSHL FLDADR  
51 52      04 78 0004B      PUSHL FLDLEN  
      00000000G0041 9F 0004F      PUSHL BUFSIZE  
66      05 FB 00056      ASHL #4, SEM_INDEX, R1  
07      50 E9 00059      PUSHAB NML$AB_PRMSEM[R1]  
B5 54      55 F2 0005C 2$:      CALLS #5, (ROUTINE_ADR)  
50      01 D0 00060      BLBC STATUS, 3$  
      04 00063 3$:      AOBLS R5, I, 1$  
      RET      #1, R0  
      : 0522  
      : 0533  
      : 0535
```

; Routine Size: 100 bytes, Routine Base: \$CODE\$ + 02B0

```
542 0536 1 ZSBTTL 'NML$DEL_FIELDS Delete parameter fields from record'
543 0537 1 GLOBAL ROUTINE NML$DEL_FIELDS (RTNDSC) =
544 0538 1
545 0539 1 ++
546 0540 1 FUNCTIONAL DESCRIPTION:
547 0541 1
548 0542 1 This routine deletes the entire list of parameters in the parameter
549 0543 1 descriptor buffer from the specified record buffer.
550 0544 1
551 0545 1 FORMAL PARAMETERS:
552 0546 1
553 0547 1 RTNDSC contains the address of the current record descriptor.
554 0548 1
555 0549 1 IMPLICIT INPUTS:
556 0550 1
557 0551 1 NML$GW_PRMDESCNT contains the number of parameter descriptors.
558 0552 1 NML$AW_PRM_DES is a list of parameter descriptors.
559 0553 1 NML$AB_PRMSEM is the parameter semantic table.
560 0554 1
561 0555 1 IMPLICIT OUTPUTS:
562 0556 1
563 0557 1 The record descriptor pointed to by RTNDSC is updated to reflect
564 0558 1 any fields deleted from the record.
565 0559 1
566 0560 1 ROUTINE VALUE:
567 0561 1 COMPLETION CODES:
568 0562 1
569 0563 1 Always returns success (NML$_STS_SUC).
570 0564 1
571 0565 1 SIDE EFFECTS:
572 0566 1
573 0567 1 NONE
574 0568 1
575 0569 1 --
576 0570 1
577 0571 2 BEGIN
578 0572 2
579 0573 2 LOCAL
580 0574 2 CPT_INDEX,
581 0575 2 SEM_INDEX,
582 0576 2 ROUTINE_ADR;
583 0577 2
584 0578 2 INCR I FROM 0 TO .NML$GW_PRMDESCNT - 1 DO
585 0579 2 BEGIN
586 0580 2
587 0581 2 CPT_INDEX = .NML$AW_PRM_DES [.I, PDB$W_INDEX];
588 0582 2 ROUTINE_ADR = .NML$AB_CPTABLE [.CPT_INDEX, CPT$A_PURGE RTN];
589 0583 2 SEM_INDEX = .NML$AB_CPTABLE [.CPT_INDEX, CPT$W_PSTINDEX];
590 0584 2
591 0585 2 (.ROUTINE_ADR) (.RTNDSC,
592 0586 2 NML$AB_PRMSEM [.SEM_INDEX, 0,0,0,0]);
593 0587 2
594 0588 2 END;
595 0589 2
596 0590 2 RETURN NML$_STS_SUC
597 0591 2
598 0592 1 END; ! End of NML$DEL_FIELDS
```


			007C 00000	.ENTRY	NML\$DEL_FIELDS, Save R2,R3,R4,R5,R6	: 0537
	55	00000000G	00 3C 00002	MOVZWL	NML\$GW_PRMDSCNT, R5	: 0578
	54		01 CE 00009	MNEGL	#1, I	: 0585
			33 11 0000C	BRB	2\$: 0581
		00000000G	0044 7F 0000E 1\$:	PUSHAQ	NML\$AW_PRM_DESC[1]	: 0582
50	53		9E 3C 00015	MOVZWL	@(SP)+, CPT_INDEX	: 0582
	53		0A C5 00018	MULL3	#10, CPT_INDEX, R0	: 0583
		00000000G	0040 9F 0001C	PUSHAB	NML\$AB_CPTABLE+6[R0]	: 0586
	56		9E D0 00023	MOVL	@(SP)+, ROUTINE_ADR	: 0578
		00000000G	0040 9F 00026	PUSHAB	NML\$AB_CPTABLE[R0]	: 0590
50	52		9E 3C 0002D	MOVZWL	@(SP)+, SEM_INDEX	: 0592
	52		04 78 00030	ASHL	#4, SEM_INDEX, R0	
		00000000G	0040 9F 00034	PUSHAB	NML\$AB_PRMSEM[R0]	
		04	AC DD 0003B	PUSHL	RTNDSC	
	66		02 FB 0003E	CALLS	#2, (ROUTINE_ADR)	
C9	54		55 F2 00041 2\$:	AOBLSS	R5, I, 1\$	
	50		01 D0 00045	MOVL	#1, R0	
			04 00048	RET		

; Routine Size: 73 bytes, Routine Base: \$CODE\$ + 0314

```
600 0593 1 XSBTTL 'NML$READPARLIST Show parameters from buffer'
601 0594 1 GLOBAL ROUTINE NML$READPARLIST (BUFDSC, MSGSIZE, TABDSC, DATDSC) =
602 0595 1
603 0596 1 ++
604 0597 1 FUNCTIONAL DESCRIPTION:
605 0598 1
606 0599 1 This routine builds a message from the list of parameters specified.
607 0600 1
608 0601 1 FORMAL PARAMETERS:
609 0602 1
610 0603 1 BUF DSC Address of message buffer descriptor.
611 0604 1 MSG SIZE Address of longword to contain resulting message size.
612 0605 1 TAB DSC Address of parameter table descriptor.
613 0606 1 DAT DSC Address of data buffer descriptor.
614 0607 1
615 0608 1 IMPLICIT INPUTS:
616 0609 1
617 0610 1 NONE
618 0611 1
619 0612 1 IMPLICIT OUTPUTS:
620 0613 1
621 0614 1 NONE
622 0615 1
623 0616 1 ROUTINE VALUE:
624 0617 1 COMPLETION CODES:
625 0618 1
626 0619 1 Always returns success (NML$_STS_SUC).
627 0620 1
628 0621 1 SIDE EFFECTS:
629 0622 1
630 0623 1 NONE
631 0624 1
632 0625 1 --
633 0626 1 BEGIN
634 0627 2
635 0628 2 MAP
636 0629 2 TAB DSC : REF DESCRIPTOR;
637 0630 2
638 0631 2 LOCAL
639 0632 2 INDEX;
640 0633 2
641 0634 2 BIND
642 0635 2 TABLE = TAB DSC [DSC$_A_POINTER] : REF BBLOCKVECTOR [, 6];
643 0636 2
644 0637 2
645 0638 2
646 0639 2 If table address is null then the specified information type is not
647 0640 2 applicable to this entity.
648 0641 2
649 0642 2 IF .TABLE EQ LA 0
650 0643 2 THEN
651 0644 2 NML$ERROR_1 (NML$_STS_FOP);
652 0645 2
653 0646 2 INCR I FROM 0 TO .TAB DSC [DSC$_W_LENGTH] - 1 DO
654 0647 2 BEGIN
655 0648 2
656 0649 2 INDEX = .TABLE [, I, 0, 0, 16, 0]; ! Get table index
```



```
: 657      0650      3
: 658      0651      3      (.TABLE [.I, 2,0,32,0]) (NML$AB PRMSEM [.INDEX, 0,0,0,0],
: 659      0652      3      .BUFDSC,
: 660      0653      3      .MSGSIZE,
: 661      0654      3      .DATDSC);
: 662      0655      3
: 663      0656      3      END;
: 664      0657      3      RETURN NML$_STS_SUC
: 665      0658      3
: 666      0659      3      END;
: 667      0660      1      ! End of NML$READPARLIST
```

54	0C	AC	003C	00000	.ENTRY	NML\$READPARLIST, Save R2,R3,R4,R5	: 0594	
			04	C1	00002	ADDL3	#4, TABDSC, R4	: 0636
			64	D5	00007	TSTL	(R4)	: 0642
			0A	12	00009	BNEQ	1\$: 0644
	00000000G	7E	0D	CE	0000B	MNEGL	#13, -(SP)	: 0646
		00	01	FB	0000E	CALLS	#1, NML\$ERROR_1	: 0652
		55	BC	3C	00015	MOVZWL	@TABDSC, R5	: 0649
		52	01	CE	00019	MNEGL	#1, I	: 0654
51		52	20	11	0001C	BRB	3\$: 0658
		51	06	C5	0001E	MULL3	#6, I, R1	: 0660
		53	64	C0	00022	ADDL2	(R4), R1	
			61	3C	00025	MOVZWL	(R1), INDEX	
			AC	DD	00028	PUSHL	DATDSC	
		7E	04	AC	7D	MOVQ	BUFDSC, -(SP)	
50		53	04	78	0002F	ASHL	#4, INDEX, R0	
	02	B1	00000000G	04	9F	PUSHAB	NML\$AB PRMSEM[R0]	
		52	04	FB	0003A	CALLS	#4, @2(R1)	
DC		50	55	F2	0003E	AOBLSS	R5, I, 2\$	
			01	D0	00042	MOVL	#1, R0	
			04	00045	RET			

; Routine Size: 70 bytes, Routine Base: \$CODE\$ + 035D

```

669 0661 1 %SBTTL 'NML$SHOWPARLIST Show parameters from QIO buffer'
670 0662 1 GLOBAL ROUTINE NML$SHOWPARLIST (BUFDSC, MSGSIZE, TABDSC, DATDSC, DATPTR) =
671 0663 1
672 0664 1 ++
673 0665 1 FUNCTIONAL DESCRIPTION:
674 0666 1
675 0667 1 This routine builds a message from the list of parameters specified.
676 0668 1
677 0669 1 FORMAL PARAMETERS:
678 0670 1
679 0671 1 BUFDSC Address of message buffer descriptor.
680 0672 1 MSGSIZE Address of longword to contain resulting message size.
681 0673 1 TABDSC Address of parameter table descriptor.
682 0674 1 DATDSC Address of data buffer descriptor.
683 0675 1 DATPTR Address of data buffer pointer.
684 0676 1
685 0677 1 IMPLICIT INPUTS:
686 0678 1
687 0679 1 NONE
688 0680 1
689 0681 1 IMPLICIT OUTPUTS:
690 0682 1
691 0683 1 NONE
692 0684 1
693 0685 1 ROUTINE VALUE:
694 0686 1 COMPLETION CODES:
695 0687 1
696 0688 1 Always returns success (NML$_STS_SUC).
697 0689 1
698 0690 1 SIDE EFFECTS:
699 0691 1
700 0692 1 NONE
701 0693 1
702 0694 1 --
703 0695 1
704 0696 2 BEGIN
705 0697 2
706 0698 2 MAP
707 0699 2 TABDSC : REF DESCRIPTOR;
708 0700 2
709 0701 2 LOCAL
710 0702 2 INDEX;
711 0703 2
712 0704 2 BIND
713 0705 2 TABLE = TABDSC [DSC$A_POINTER] : REF BBLOCKVECTOR [, 6];
714 0706 2
715 0707 2 INCR I FROM 0 TO .TABDSC [DSC$W_LENGTH] - 1 DO
716 0708 2 BEGIN
717 0709 2
718 0710 2 INDEX = .TABLE [.I, 0,0,16,0]; ! Get table index
719 0711 2
720 0712 2 (.TABLE [.I, 2,0,32,0]) (NML$AB_PRMSSEM [.INDEX, 0,0,0,0],
721 0713 2 .BUFDSC,
722 0714 2 .MSGSIZE,
723 0715 2 .DATDSC,
724 0716 2 .DATPTR);
725 0717 3
```



```

: 726      0718 2      END;
: 727      0719 2
: 728      0720 2      RETURN NML$_STS_SUC
: 729      0721 2
: 730      0722 1      END;
! End of NML$SHOWPARLIST

```

55	0C	AC	04	003C	00000	.ENTRY	NML\$SHOWPARLIST, Save R2,R3,R4,R5	: 0662
		54	BC	C1	00002	ADDL3	#4, TABDSC, R5	: 0705
		52	0C	3C	00007	MOVZWL	@TABDSC, R4	: 0707
			01	CE	0000B	MNEGL	#1, I	: 0713
51		52	21	11	0000E	BRB	2\$: 0710
		51	06	C5	00010	MULL3	#6, I, R1	
		53	65	C0	00014	ADDL2	(R5), R1	
		7E	61	3C	00017	MOVZWL	(R1), INDEX	
		7E	10	AC	7D	MOVQ	DATDSC, -(SP)	: 0715
50		7E	04	AC	7D	MOVQ	BUFDSC, -(SP)	: 0713
		53	04	78	00022	ASHL	#4, INDEX, R0	: 0712
	02	B1	00000000G00	40	9F	PUSHAB	NML\$AB PRMSEM[R0]	
DB		52	05	FB	0002D	CALLS	#5, @2(R1)	: 0707
		50	54	F2	00031	AOBLSS	R4, I, 1\$: 0720
			01	D0	00035	MOVL	#1, R0	: 0722
			04	00038	RET			

; Routine Size: 57 bytes, Routine Base: \$CODE\$ + 03A3

```
732 0723 1 $SBTTL 'NML$BLDALLDES Build parameter descriptors from record'
733 0724 1 GLOBAL ROUTINE NML$BLDALLDES (RECDSC, TABDSC) =
734 0725 1
735 0726 1 ++
736 0727 1 FUNCTIONAL DESCRIPTION:
737 0728 1
738 0729 1 This routine is used by SET ALL functions to build parameter
739 0730 1 descriptors from a permanent data base record.
740 0731 1
741 0732 1 FORMAL PARAMETERS:
742 0733 1
743 0734 1 RECDSC Address of the current record descriptor.
744 0735 1 TABDSC Address of parameter table descriptor.
745 0736 1
746 0737 1 IMPLICIT INPUTS:
747 0738 1
748 0739 1 NML$AB_PRMSEM is the parameter semantic table.
749 0740 1
750 0741 1 IMPLICIT OUTPUTS:
751 0742 1
752 0743 1 NONE
753 0744 1
754 0745 1 ROUTINE VALUE:
755 0746 1 COMPLETION CODES:
756 0747 1
757 0748 1 Always returns success (NML$_STS_SUC).
758 0749 1
759 0750 1 SIDE EFFECTS:
760 0751 1
761 0752 1 NONE
762 0753 1
763 0754 1 --
764 0755 1
765 0756 2 BEGIN
766 0757 2
767 0758 2 MAP
768 0759 2 RECDSC : REF DESCRIPTOR,
769 0760 2 TABDSC : REF DESCRIPTOR;
770 0761 2
771 0762 2 LOCAL
772 0763 2 FLDADR,
773 0764 2 FLDSIZE,
774 0765 2 INDEX;
775 0766 2
776 0767 2 BIND
777 0768 2 TABLE = TABDSC [DSC$A_POINTER] : REF BLOCK;
778 0769 2
779 0770 2 NML$GW_PRMDESCNT = 0; ! Reset parameter descriptor count
780 0771 2
781 0772 2 INCR I FROM 0 TO .TABDSC [DSC$W_LENGTH] - 1 DO
782 0773 2 BEGIN
783 0774 2
784 0775 2 FLDADR = 0;
785 0776 2
786 0777 2 IF NML$SEARCHFLD (.RECDSC,
787 0778 2 .TABLE [.I,0,16,0],
788 0779 2 FLDSIZE,
```



```
: 789      0780 3      FLDADR)
: 790      0781 3      THEN
: 791      0782 4      BEGIN
: 792      0783 4
: 793      0784 4      INDEX = .TABLE [.I,16,16,0];
: 794      0785 4
: 795      0786 4      NML$SAVEPARAM (.INDEX,
: 796      0787 4      .FLDSIZE,
: 797      0788 4      .FLDADR);
: 798      0789 3      END;
: 799      0790 2      END;
800      0791 2      RETURN NML$_STS_SUC
801      0792 2
802      0793 2
: 803      0794 1      END;

! End of NML$BLDALLDES
```

				003C 00000	.ENTRY NML\$BLDALLDES, Save R2,R3,R4,R5	: 0724
				08 C2 00002	SUBL2 #8, SP	
53	08	5E		04 C1 00005	ADDL3 #4, TABDSC, R3	: 0768
		AC	00000000G	00 B4 0000A	CLRW NML\$GW PRMDESCNT	: 0770
		54	08	BC 3C 00010	MOVZWL @TABDSC, R4	: 0772
		52		01 CE 00014	MNEGL #1, I	: 0777
				30 11 00017	BRB 2\$	
				6E D4 00019 1\$:	CLRL FLDADR	: 0775
				5E DD 0001B	PUSHL SP	: 0777
			08	AE 9F 0001D	PUSHAB FLDSIZE	
			00 B342	DF 00020	PUSHAL @0(R3)[1]	: 0778
		7E		9E 3C 00024	MOVZWL @0(SP)+, -(SP)	
			04	AC DD 00027	PUSHL RECDSC	: 0777
		00000000G	00	04 FB 0002A	CALLS #4, NML\$SEARCHFLD	
		15		50 E9 00031	BLBC R0, 2\$	
55	9E	10	00 B342	DF 00034	PUSHAL @0(R3)[1]	: 0784
			10	EF 00038	EXTZV #16, #16, @0(SP)+, INDEX	
			08	6E DD 0003D	PUSHL FLDADR	: 0788
				AE DD 0003F	PUSHL FLDSIZE	: 0787
				55 DD 00042	PUSHL INDEX	: 0786
	FBDB	CF		03 FB 00044	CALLS #3, NML\$SAVEPARAM	
CC		52		54 F2 00049 2\$:	AOBLSS R4, I, 1\$: 0772
		50		01 D0 0004D	MOVL #1, R0	: 0792
				04 00050	RET	: 0794

; Routine Size: 81 bytes, Routine Base: \$CODE\$ + 03DC

NML\$PMANIP
V04-000

NML internal parameter manipulation module
NML\$BLDALLDES Build parameter descriptors from

N 13
16-Sep-1984 00:26:09
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLPMANIP.B32;1

Page 26
(11)

: 805 0795 1 END
: 806 0796 1
: 807 0797 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name Bytes Attributes
\$CODE\$ 1069 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
\$255\$DUA28:[NML.OBJ]NMLLIB.L32;1	341	38	11	27	00:00.1
\$255\$DUA28:[SHRLIB]NMLIBRY.L32;1	887	3	0	47	00:00.2
\$255\$DUA28:[SHRLIB]NET.L32;1	1279	14	1	63	00:00.3
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	5	0	581	00:03.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:NMLPMANIP/OBJ=OBJ\$:NMLPMANIP MSRC\$:NMLPMANIP/UPDATE=(ENH\$:NMLPMANIP)

: Size: 1069 code + 0 data bytes
: Run Time: 00:24.0
: Elapsed Time: 01:02.4
: Lines/CPU Min: 1991
: Lexemes/CPU-Min: 12432
: Memory Used: 169 pages
: Compilation Complete

0285 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

